COMMUNITY DETECTION IN COMPLEX NETWORK: METRIC SPACE, NEAREST NEIGHBOR SEARCH, LOW-RANK APPROXIMATION AND OPTIMALITY

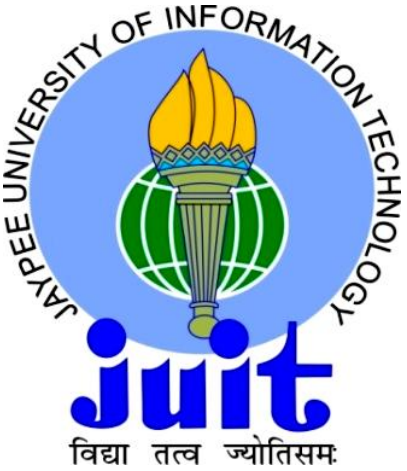
Synopsis submitted in fulfillment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY

By

SUMAN SAHA

Enrollment No. 136212



Under the supervision of

Prof. S. P. Ghrera

Department of Computer Science and Engineering

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

April, 2016

ABSTRACT

Community detection in complex network is an important problem of much interest in recent years. In general, a community detection algorithm chooses an objective function and captures the communities of the network by optimizing the objective function and then one uses various heuristics to solve the optimization problem to extract the interesting communities for the user.

The objective of this thesis is to bridge the gap between two important research directions, 1) complex network analysis, which deals with large real graphs but generally studied via graph theoretic analysis or spectral analysis and 2) data analysis in metric space using geometric distances, nearest neighbor search, which is a fundamental computational tool for large data analysis and low-rank approximation of distance matrix. In the second chapter, we have demonstrated the procedure to transform a graph into points of a metric space and developed the methods of community detection with the help of metric defined for pair of points. We have also studied and analyzed the community structure of the network therein.

The results obtained with our approach are very competitive with most of the well known algorithms in the literature and this is justified over the large collection of datasets. On the other-hand, it can be observed that time taken by our algorithm is quite less compared to other methods. In the third chapter, we have studied the nearest neighbor search problem in complex network by the development of a suitable notion of nearness. The computation of efficient nearest neighbor search among the nodes of a complex network using the metric tree and locality sensitive hashing (LSH) are also studied and experimented.

For evaluation of the proposed nearest neighbor search on complex network, we applied it in network community detection problem. Experiments are performed to verify the usefulness of nearness measures for the complex networks and the role of metric tree and LSH to compute fast community detection using nearest neighbor search. We observed that nearest neighbor between network nodes is a very efficient tool to explore better community structure of the real networks, several efficient approximation scheme are very useful for large networks, which hardly made any degradation of results, whereas saves lot of computational times.

In the fourth chapter, we explored two low rank approximation methods, Nystrom sampling and random sketching. The Nystrom methods approximate any symmetric positive semi-definite matrix in terms of a subset of its columns and achieve the low complexity. Random sketching is described by the use of Johnson-Lindenstrauss theorem, which ensures the preservation of distance during lower dimensional approximation and provide minimum lower bound of approximation. Achievements of low rank methods are fast computation on massive networks and approximated results with moderate lower bound.

Finally, in the fifth chapter, we combine the communities of complex networks determined by several algorithms with Rough Sets theory to obtain a new optimal community structure of the network. We obtained improved results of community detection in less time by combining results of approximate algorithms.

Table of Contents

	PAGE NO.
ABSTRACT	ii-iii
1. INTRODUCTION	1-4
1.1 Complex Network	1
1.2 Network community detection & related literatures	2
1.3 Real networks	4
1.4 Objective of the thesis	4
2. NETWORK COMMUNITY DETECTION ON METRIC SPACE	5-11
2.1 Network to metric space	5
2.2 Community detection on induced metric space	8
2.3 Experiments on metric based community detection	10
2.4 Achievements	11
3. NEAREST NEIGHBOR SEARCH FOR NETWORK COMMUNITY DETECTION	12-17
3.1 Nearest Neighbor search in Complex Network	12
3.2 Nearest Neighbor search using Metric-tree	13
3.3 Nearest neighbor search using locality sensitive hashing	14
3.4 Experiments on nearest neighbor search	15
3.5 Achievements	17
4. LOW-RANK APPROXIMATION FOR VERY LARGE NETWORKS	18-20
4.1 Low-rank approximation methods	18
4.2 Experiments on low-rank approximation	19
4.3 Achievements	20
5. OPTIMAL ESTIMATION FOR COMMUNITY DETECTION	21-23
5.1 Theoretical developments	21
5.2 Experiments on optimal communities	22
5.3 Achievements	23
6. CONCLUSION	24-24
7. REFERENCES	25-30
8. LIST OF PUBLICATIONS	31-31

1. INTRODUCTION

Complex network analysis is a large and growing body of research on the measurement and analysis of relational structure. Issues pertaining to data collection, analysis of single networks, network comparison, and analysis of individual-level covariates are discussed, and a number of suggestions are made for avoiding common pitfalls in the application of network methods to substantive questions. The complex network field is an interdisciplinary research program which seeks to predict the structure of relationships among social and natural entities, as well as the impact of said structure on other natural phenomena.

1.1. Complex Network

A complex network is simply a structure consisting of entities with a relationship among those entities that represents interaction, collaboration or influence between entities (relationship: friendship, an email sent, the co-authorship, and so forth). Mathematically, a complex network is a graph G = (V, E), where the nodes represent entities, and an edge (u, v) ∈ E denotes some type of relationship between the entities u and v.

Researchers have concentrated particularly on a few properties that seem to be common to many networks: the small-world property, power-law degree distributions, and high clustering coefficient. Small world effect is the finding that the average distance between vertices in a network is short, usually scaling logarithmically with the total number n of vertices. Right-skewed degree distribution is another property that many networks possess. The degree of a vertex in a network is the number of other vertices to which it is connected, and one finds that there are typically many vertices in a network with low degree and a small number with high degree, the precise distribution follow a power-law or exponential form. The clustering coefficient of complex networks is much higher than is predicted in a random graph. Informally, the clustering coefficient measures the probability that two people who have a common friend will themselves be friends or not, in graph theoretic terms, the fraction of triangles in the graph that are closed.

A large number of examples are available for real life complex networks like, citation graphs: (directional) edge (X,Y) exists if paper X cites paper Y, collaboration graphs: (bidirectional) edge (X,Y) if person X worked with person Y, semantic graphs: dictionaries, thesaurus; edge (X,Y) exists if word X is associated with word Y, biological graphs: edge (X,Y) exists if process X is related to process Y (e.g. protein interactions, predators, food webs), communication graphs: computer networks or telephone networks, news graphs: relationship of events, words, or people in the news, the Internet and the worldwide web is a directed graph and social networks (facebook, twitter, Google+,...) to name a few.

1.2.Network community detection & related literatures

Community detection in a network aims to identify the modules and their hierarchical organization using link structure [39, 53]. In general the network has to be sparse, there is no meaning of communities in a dense network [47]. Community detection algorithms are NP-hard [12] to solve. Several example of communities are social communities in social network, protein group having same specification in protein interaction network and related topics in world wide web to name a few.

1.2.1. Performance indicator

Several measures for quantifying the quality of communities have been pro- posed, they mostly consider that communities are set of nodes with many edges between them and few connections with nodes of different communities. Some of the community evaluation measures are described below. There are several performance indicators for community detection algorithms.

Modularity: The notion of modularity is the most popular for the network community detection purpose. The modularity index assigns high scores to communities whose internal edges are more than that expected in a random-network model which preserves the degree distribution of the given network.

Conductance: Conductance is widely used for graph partitioning literature. The conductance of a set S with complement S^C is the ratio of the number of edges connecting nodes in S to nodes in S^C by the total number of edges incident to S or to S^C (whichever number is smaller).

1.2.2. Popular algorithms

In this paragraph we have given a brief list of the algorithms developed for network community detection purposes. The basic approach and the complexity of execution are also briefed in the table below 1.

Fast Greedy Algorithm: This algorithm [11, 40] is modularity-based and uses a hierarchical agglomerative approach. It is called fast greedy because it is significantly faster than older algorithms and use greedy method.

Walktrap Algorithm: This algorithm by Pons and Latapy [46] uses a hierarchical agglomerative method. Here, the distance between two nodes is defined in terms of random walk process. The basic idea is that if two nodes are in the same community, the probability to get to a third node located in the same community through a random walk should not be very different. The distance is constructed by summing these differences over all nodes, with a correction for degree.

Eigenvector Algorithm: This algorithm by Newman [38] is modularity-based, and it uses an optimization method inspired by graph partitioning techniques. It relies on the eigenvectors of a so-called modularity matrix, instead of the graph Laplacian traditionally used in graph partitioning.

Label Propagation Algorithm: This algorithm by Raghavan et al. [48] uses the concept of node neighborhood and the diffusion of information in the network to identify communities. Initially, each node is labeled with a unique value. Then an iterative process takes place, where each node takes the label which is the most spread in its neighborhood. This process goes on until one of several conditions is met, for in-stance no label change. The resulting communities are defined by the last label values.

Spinglass Algorithm: This algorithm by Reichardt and Bornholdt [49] is an optimization method relying on an analogy between the statistical mechanics of complex networks and physical spin glass models. There are more algorithms developed to solve network community detection problem a complete list can be obtained in several survey articles [20, 35, and 57]. Some interesting recent articles are [1, 4, 15, 16, 33, 42, and 58].

Table 1: Algorithms for network community detection and their complexities

Author	Ref.	Cat.(No.)	Order
Van Dongen	(Graph clustering, 2000 [55])	GT(1)	$O(nk^2)$, $k < n$ parameter
Eckmann & Moses	(Curvature, 2002 [19])	GT(2)	$O(mk^2)$
Girvan & Newman	(Modularity, 2002 [24])	SDP(1)	$O(n^2m)$
Zhou & Lipowsky	(Vertex Proximity, 2004 [59])	GT(3)	$O(n^3)$
Reichardt & Bornholdt	(spinglass, 2004 [49])	SDP(2)	parameter dependent
Clauset et al.	(fast greedy, 2004 [11])	SDP(3)	$O(nlog_2n)$
Newman & Girvan	(eigenvector, 2004 [39])	SP(1)	$O(nm^2)$
Wu & Huberman	(linear time, 2004 [56])	GT(4)	$O(n+m)$
Fortunato et al.	(infocentrality, 2004 [21])	SDP	$O(m^3n)$
Radicchi et al.	(Radicchi et al., 2004 [47])	SP(2)	$O(m^4/n^2)$
Guimera et al.	(Simulated Annealing, 2004 [26])	SDP(5)	parameter dependent
Capocci et al.	(Capocci et al., 2004 [7])	SP(3)	$O(n^2)$
Latapy & Pons	(walktrap, 2004 [46])	SP(4)	$O(n^3)$
Duch & Arenas	(Extremal Optimization, 2005 [18])	GT(5)	$O(n^2logn)$
Bagrow & Bollt	(Local method, 2005 [6])	SDP(6)	$O(n^3)$
Palla et al.	(overlapping community, 2005 [41])	GT(6)	$O(exp(n))$
Raghavan et al.	(label propagation, 2007 [48])	GT(7)	$O(n+m)$
Rosvall & Bergstrom	(Infomap, 2008 [51])	SP(5)	$O(m)$
Ronhovde & Nussinov	(Multiresolution community, 2009 [50])	GT(8)	$O(m\beta logn)$, $\beta \approx 1.3$
De Meo et al.	(Mixing information, 2014 [15])	SDP	$O(n^3)$

1.3.Real networks

A list of real networks taken from several real life interactions is considered for our experiments and they are shown in Table 2 below. We have also listed the number of nodes, number of edges, average diameter, and the number of communities (k).

Table 2: Complex network datasets and values of their parameters

Name	Type	# Nodes	# Edges	Diameter	k
DBLP	U	317,080	1,049,866	8	268
Arxiv-AstroPh	U	18,772	396,160	5	23
web-Stanford	D	281,903	2,312,497	9.7	69
Facebook	U	4,039	88,234	4.7	164
Gplus	D	107,614	13,673,453	3	457
Twitter	D	81,306	1,768,149	4.5	213
Epinions1	D	75,879	508,837	5	128
LiveJournal1	D	4,847,571	68,993,773	6.5	117
Orkut	U	3,072,441	117,185,083	4.8	756
Youtube	U	1,134,890	2,987,624	6.5	811
Pokec	D	1,632,803	30,622,564	5.2	246
Slashdot0811	D	77,360	905,468	4.7	81
Slashdot0922	D	82,168	948,464	4.7	87
Friendster	U	65,608,366	1,806,067,135	5.8	833
Amazon0601	D	403,394	3,387,388	7.6	92
P2P-Gnutella31	D	62,586	147,892	6.5	35
RoadNet-CA	U	1,965,206	5,533,214	500	322
Wiki-Vote	D	7,115	103,689	3.8	21

1.4. Objective of the thesis

The objective of this thesis is to bridge the gap between two important research directions, 1) complex network analysis, which deals with large real graphs but generally studied via graph theoretic analysis or spectral analysis and 2) data analysis in metric space using geometric distances, nearest neighbor search, which is a fundamental computational tool for large data analysis and low-rank approximation of distance matrix. Network community detection is not easy NP-Hard like data clustering due to the lack of good heuristics. Both, Graph traversal based methods and spectral methods are computationally overloaded due to the verification of objective function value, required to guide next iteration. Rich literature of clustering is not suitable for graph data. These observations motivate us for a transformation from complex network to Metric Space. But the metrics developed so far on graph (like Shortest path, Jaccard similarity and Euclidean distance between adjacency vectors) are less successful for network community detection in terms of conductance and modularity so there is a need for development of a good metric which works better on complex network. Aim of this work is the development of fast and accurate algorithms for community detection of large network.

Competitiveness of the several approximation methods are analyzed with respect to their modularity value and computational complexity. The main contribution of the thesis is explained via the following chapters: 1) Chapter 2: Network community detection on metric space 2) Chapter 3: Nearest Neighbor search in Complex Network for Community Detection, 3) Chapter 4: Low rank approximations for community detection of very large networks, and 4) Chapter 5: Optimal evaluation of network community.

2. Network Community Detection on Metric Space

We develop the notion of a metric among the nodes using some new matrices derived from the modified adjacency matrix of the graph. The developed metric space is flexible over the networks and can be tuned to enhance its community structure. We also proposed the community detection algorithms on induced metric space and analyze the results and complexities of the developed algorithms.

2.1. Network to metric space

In this subsection we have demonstrated the algorithm to convert the nodes of the graph to the points of a metric space preserving the community structure of the graph. The algorithm depends on the sub modules 1) construction of Lx (L1 or L2) and 2) obtaining a structure preserving distance function. The algorithm works as picking pair of nodes from Lx and computing distance defined in the second module.

2.2 L_z construction

The L1 is defined as $L1 = D(\lambda) + A$, where A is the adjacency matrix of the given network and $D(\lambda)$ is a diagonal matrix of same size with diagonal values equal to a non negative constant λ. The L2 is defined as $L2 = D(\lambda x) + A$, where A is the adjacency matrix of the given network and $D(\lambda x)$ is a diagonal matrix of same size with diagonal values determine by a non negative function λx of the node x. The choice of λ and λx plays a crucial role in combination with the function chosen in the second module for determination of a suitable metric and is discussed later part of this subsection.

2.3 Function selection

The function selection module determines the metric for pair of nodes. The function selector φ converts a similarity function (Pearson coefficient (σ_{PC}), Spacerman coefficient (σ_{SC}) or Cosine similarity (σ_{CS}) into a distance matrix. In general the similarity function satisfies the positivity and similarity condition of metric but not triangle inequality. φ is a metric

preserving (φ(d(xi, xj) = dφ(xi, xj)), concave and monotonically increasing function. The three conditions above refer to as chord condition. The φ function is chosen to have minimum internal area with chord.

2.4 Choice of λ and φ(σ)()

The choices in the above sub modules play a crucial role in the graph to metric transformation algorithm to be used for community detection. The complex network is characterized by small average diameter and high clustering coefficient. Several studies on network structure analysis reveal that there are hub nodes and local nodes characterizing the interesting structure of the complex network. Suppose we have taken φ = arccos, σ_{CS} and constant $\lambda \geq 0$. $\lambda = 0$ penalize the effect of direct edge in the metric and is suitable to extract communities from highly dense graph. $\lambda = 1$ place the similar weight of direct edge and common neighbor reduce the effect of direct edge in the metric and is suitable to extract communities from moderately dense graph. $\lambda = 2$ set more importance to direct edge than common neighbor (this is the common case of available real networks). $\lambda \geq 2$ penalize the effect of common neighbor in the metric and is suitable to extract communities from very sparse graph. The algorithm for transforming a graph to the points of a metric space is given below 5.

Algorithm 1 Mapping a graph into Metric space

Require: $G = (V, E)$
Ensure: $M = (V, d)$
1: $D^{\lambda}_{(n\times n)} = \begin{cases} 0 \ if \ i \neq j \\ \lambda \geq 0 \ if \ i = j \end{cases}$
2: $A = D^{\lambda} + E$
3: **for** $i = 1$ to n **do**
4: **for** $j = 1$ to n **do**
5: $d(v_i, v_j) = \phi(1 - \frac{a_i \cdot a_j}{|a_i||a_j|})$, where $v_i, v_j \in V$ and a_k is the k^{th} row of A and ϕ is an affine function.
6: **end for**
7: **end for**
8: **return** $M = (V, d)$

Theorem 2.1: M = (V, d) constructed in the above algorithm 5 is a metric space with respect to the metric d, i.e.

2.5 Community detection on induced metric space

In this section we have explore k partitioning algorithm for the purpose of network community detection by using the metric space constructed above for each graph. We have also studied and analyzed the advantages of the k partitioning method over the standard algorithm for network community detection.

2.5.1 k-partitioning

The community detection methods based on K-partitioning of graph is possible using newly defined node distance, because the nodes of the graph are converted into the points of a metric space. The k-partitioning of graph using this distance converges automatically and does not compute the value of objective function in iterations therefore reduce the computation compared to standard graph partitioning methods. The results of k-partitioning of graph using metric are competitive on a large collection of networks. The algorithm for community detection using k-partitioning and its details analysis is given below. Before that we need to determine the value of k and is discussed in the next subsection.

2.5.2. k selection

Determining the optimal number of k is an important problem for community detection researchers. An extensive analysis can be found in the work of Leskovec et al. [34]. The standard practice is to solve an optimization equation with respect to k for which the optimal value of the objective function is achieved. Another method based on farthest first traversal is also very useful in terms of computational efficiency [25]. For small networks the global optimization works better and for very large network the second choice give the faster approximate solution.

Algorithm 2 k-center partitioning algorithm

Require: $M = (V, d)$
Ensure: $T = \{C_1, C_2, \ldots, C_k\}$ with minimum $cost(T)$
1: Initialize centers $z_1, \ldots, z_k \in R^n$ and clusters $T = \{C_1, C_2, \ldots, C_k\}$
2: **repeat**
3: **for** $i = 1$ to k **do**
4: **for** $j = 1$ to k **do**
5: $C_i \leftarrow \{x \in V \ s.t. \ |z_i - x| \leq |z_j - x|\}$
6: **end for**
7: **end for**
8: **for** $j = 1$ to k **do**
9: $z_i \leftarrow mean(C_i)$
10: **end for**
11: **until** $|cost(T_t) - cost(T_{t+1})| = 0$
12: **return** $T = \{C_1, C_2, \ldots, C_k\}$

2.6 Convergence

Convergence of the network community detection algorithms are the least studied research areas of network science. However, the rate of convergence is one of the important issues and low rate of convergence is the major pitfall of the most of the existing algorithms. Due to the transformation into the metric space, our algorithm equipped with the quick convergence facility of the k-partitioning on metric space by providing a good set of initial points. Another crucial pitfall suffer by majority of the existing algorithms is the validation of the objective function in each iteration during convergence. Our algorithm converges automatically to the optimal partition thus reduces the cost of validation during convergence.

Theorem 2.2: During the course of the k center partitioning algorithm, the cost monotonically decreases.

Theorem 2.3: If T is the solution returned by farthest-first traversal, and T^o is the optimal solution, then $cost(T^o) \leq cost(T) \leq 2cost(T^o)$

2.7 Experiments on metric based community detection

We performed many of experiments to test the proposed network detection method via induced metric space over several real networks2. Objective of the experiment is to verify behavior of the algorithm and the time required to compute the algorithm. In this experiment we have compared two groups of algorithms for network community detection with our proposed algorithm based on metric space. Experiment is performed on a large list of network data sets (table 2). Three groups of algorithms are considered; first group contain algorithms based on spectral algorithms, the second group contains algorithms based on semi definite programming and the third group contain algorithms based on graph traversal approaches. For each group we have taken the best value of modularity in table 3 among all the algorithms in the groups. The results obtained with our approach are very competitive with most of the well known algorithms in the literature and this is justified over the large collection of datasets.

Table 3: Comparison of our approaches with other best methods in terms of modularity, the number inside the brackets denotes the algorithm of the group

Name	Spectral	SDP	GT	Metric
Facebook	0.4487(1)	0.5464(4)	0.5434(5)	0.5472
Gplus	0.2573(1)	0.4047(3)	0.3998(5)	0.4056
Twitter	0.3261(3)	0.3706(1)	0.3691(7)	0.3709
Epinions1	0.0280(1)	0.1440(3)	0.1401(5)	0.1447
LiveJournal1	0.0791(1)	0.1455(5)	0.1432(5)	0.1458
Pokec	0.0129(3)	0.0294(1)	0.0288(5)	0.0295
Slashdot0811	0.0038(1)	0.0130(4)	0.0127(7)	0.0131
Slashdot0922	0.0045(1)	0.0176(5)	0.0171(5)	0.0176
Friendster	0.0275(4)	0.0536(5)	0.0526(7)	0.0536
Orkut	0.0294(3)	0.0689(4)	0.0675(5)	0.0690
Youtube	0.0096(1)	0.0934(2)	0.0903(5)	0.0936
DBLP	0.4011(5)	0.4214(1)	0.4207(5)	0.4215
Arxiv-AstroPh	0.4174(3)	0.5079(3)	0.5045(5)	0.5081
web-Stanford	0.3595(5)	0.3908(4)	0.3896(7)	0.3908
Amazon0601	0.1768(1)	0.2649(4)	0.2615(7)	0.2650
P2P-Gnutella31	0.0009(1)	0.0522(2)	0.0503(5)	0.0523
RoadNet-CA	0.0212(3)	0.1690(4)	0.1633(5)	0.1692
Wiki-Vote	0.0266(1)	0.2093(1)	0.2023(5)	0.2095

Table 4: Comparison of our approaches with other best methods in terms of time

Algorithm	Spectral	SDP	GT	Metric
Minimum Time	884	910	871	869
Maximum Time	1386	1725	1641	869
Average Time	917	981	1338	869

2.8 Achievements

The main achievement of the work was to use the rich literature of clustering in metric space. Clustering is easy NP-Hard in metric space, whereas network community detection is NP-Hard. The results obtained with our approach were very competitive with most of the well known algorithms in the literature and justified over the large collection of datasets. Our algorithm converges automatically to optimal clustering. Verification of objective function is not required to determine next iteration, like popular approaches do, thus saving the time of computation.

3 Nearest neighbor search for network community detection

The main contributions of this work are 1) development of the concept of nearness between the nodes of a complex network, 2) comparing the proposed nearness with other notions of similarities, 3) study and experiment on approximate nearest neighbor search for complex network using M-tree and LSH, 4) design of efficient community detection algorithm using nearest neighbor search. We observed that nearest neighbor between network nodes is a very efficient tool to explore better community structure of the real networks. Further several efficient approximation scheme are very useful for large networks, which hardly made any degradation of results, whereas saves lot of computational times.

3.1 Nearest Neighbor search in Complex Network

The nearest neighbor (NN) search is a fundamental computational primitive for structural analysis of data and other query retrieval purposes. NN search is very useful for dealing with massive data sets, but it suffers with curse of dimension [52, 54]. A recent surge of results shows that it is most useful tool for analyzing very large data, provided a suitable space partitioning data structure is used, like, kd-tree, quad-tree, R-tree, metric-tree and locality sensitive hashing [13,22,28,43]. One more advantage of using nearest neighbor search for large data analysis is the availability of efficient approximation scheme, which provides almost same results in very less time [3, 36].

There are numerous methods developed to compute the nearest neighbor search for points of a metric space. However, finding the nearest neighbor search on some data where dimension is high suffer from curse of dimension. Some recent research on this direction revealed that dimension constrained can be tackled by using efficient data structures like metric tree and locality sensitive hashing. In this section we have explored metric tree to perform the nearest neighbor search on complex network with the help of metric mapping of complex network described in the previous section.

3.2 Nearest Neighbor search using Metric-tree

A metric tree is a data structure specially designed to perform the nearest neighbor query for the points residing on a metric space and perform well on high dimension particularly when some approximation is permitted. A metric tree organizes a set of points in a spatial hierarchical manner. It is a binary tree whose nodes represent a set of points. The root node represents all points, and the points represented by an internal node v are partitioned into two subsets, represented by its two children. Formally, if we use N (v) to denote the set of points represented by node v, and use v.lc and v.rc to denote the left child and the right child of node v, then we have N (v) = N (v.lc) ∪ N (v.rc) φ = N (v.lc) ∩ N (v.rc) for all the non-leaf nodes. At the lowest level, each leaf node contains very few points.

An M-Tree [9] consists of leaf node, internal node and routing object. Leaf nodes are set of objects N_v with pointer to parent object v_p. Internal nodes are set of routing objects NRO with pointer to its parent object v_p. Routing object vr store covering radius $r(v_r)$ and pointer to covering tree $T(v_r)$, Distance of vr from its parent object $d(v_r, P(v_r))$. Feature values stored in the object v_j are object identifier oid (v_j) and distance of v_j from its parent object $d(v_j, P(v_j))$

The key to building a metric-tree is how to partition a node v. A typical way is as follows: We first choose two pivot points from N (v), denoted as v.lpv and v.rpv. Ideally, v.lpv and v.rpv are chosen so that the distance between them is the largest of all distances within N (v). More specifically, $||v.lpv - v.rpv|| = \max_{p1, p2 \in N(v)} ||p1 - p2||$. A search on a metric-tree is performed using a stack. The current radius r is used to decide which child node to search first. If the query q is on the left of current point, then v.lc is searched first, otherwise, v.rc is searched first. At all times, the algorithm maintains a candidate NN and distance between them determines the current radius, which is the nearest neighbor it finds so far while traversing the tree. The algorithm for nearest neighbor search using metric tree is given below 3.

Algorithm 3 NN search in M-Tree

Require: $M = (V, d)$ & q
Ensure: $d(q, v_q)$
1: Insert root object v_r in stack
2: Set current radius as $d(v_r, q)$
3: Successively traverse the tree in search of q
4: PUSH all the objects of traversal path into stack
5: Update the current radius
6: If leaf object reached
7: POP objects from stack
8: For all points lying inside the ball of current radius centering q, verify for possible nearest neighbor and update the current radius.
9: **return** $d(q, v_q)$

The theoretical advantage of using metric tree as a data structure for nearest neighbor search is: Let M = (V, d), be a bounded metric space. Then for any fixed data $V \in R^n$ of size n, and for constant $c \geq 1$, $\exists$ such that we may compute $d(q, V)| \in$ with at most $c \bullet [\log(n) + 1]$ expected metric evaluations [10].

3.3 Nearest neighbor search using locality sensitive hashing

Metric trees, so far represent the practical state of the art for achieving efficiency in the largest dimension possible. However, many real-world problems consist of very large dimension and beyond the capability of such search structures to achieve sub-linear efficiency. Thus, the high-dimensional case is the long-standing frontier of the nearest-neighbor problem. The approximate nearest neighbor can be computed very efficiently using Locality sensitive hashing. Several methods to compute first nearest neighbor query exist in the literature and locality-sensitive hashing (LSH) is most popular because of its dimension independent run time [37, 44]. In a locality sensitive hashing, the hash function has the property that close points are hash into same bucket with high probability and distance points are hash into same bucket with low probability. Mathematically, a family $H = \{h: S \rightarrow U\}$ is called (r1, r2, p1, p2)-sensitive if for any $p, q \in S$

- if $p \in B(q, r1)$ then $Pr_H[h(q) = h(p)] \geq p1$
- if $p \notin B(q, r2)$ then $Pr_H[h(q) = h(p)] \leq p2$

where B(q, r) denotes a hyper sphere of radius r centered at q. In order for a locality-sensitive family to be useful, it has to satisfy inequalities $p_1 > p_2$ and $r_1 < r_2$ when D is a distance, or $p_1 > p_2$

and $r_1 > r_2$ when D is a similarity measure [23,29]. The value of $\delta = \log(1/P_1)/\log(1/P_2)$ determines search performance of LSH. Defining a LSH as a(r, r(1 + €), p1, p2), the (1 + €) NN problem can be solved via series of hashing and searching within the buckets [14,22,31].

In this paragraph, we discuss the existence of locality sensitive hash function families for the proposed metric for complex network. The LSH data structure stores all nodes in hash tables and searches for nearest neighbor via retrieval. The hash table is contain many buckets and identified by bucket id. Unlike conventional hashing, the LSH approach tries to maximize the probability of collision of near items and put them into same bucket. For any given the query q the bucket h(q) considered to search the nearest node. In general k hash functions are chosen independently and uniformly at random from hash family H. The output of the nearest neighbor query is provided from the union ok k buckets. The consensus of k functions reduces the error of approximation. We considered k random points from the metric space. Each random point r_i define a hash function hi(x) = sign (d(x, r_i)), where d is the metric and i ∈ [1, k]. These randomized hash functions are locality sensitive [5, 8].

Algorithm 4 NN search in LSH

Require: $M = (V, d)$ & q

Ensure: $d(q, V)$

1: Identify buckets of query point q corresponding to different hash functions.
2: Compute nearest neighbor of q only for the points inside the selected buckets.
3: **return** $d(q, V)$

The theoretical advantage of using locality sensitive hashing as a data structure for nearest neighbor search is: Let M = (V, d), be a bounded metric space. Then for any fixed data $V \in R^n$ of size n, and for constant $c \geq 1$, ∃ € such that we may compute d (q, V)|€ with at most $mn^{O(1/€)}$ expected metric evaluations, where m is the number of dimension of the metric space. In case of complex network m = n so expected time is $n^{O(2/€)}$ [10, 27].

3.4 Experiments on nearest neighbor search

In this subsection we described in details several experiments to assess the, proposed nearness measure for the nodes of the network, efficiency of several approximation scheme to compute node nearness and performance of proposed algorithm for community detection. We performed

three different experiments to assess the performance of the proposed network nearest neighbor search for community detection. First experiment is designed to evaluate the nearness measure, second experiment is designed to explore the effectiveness of approximate nearest neighbor search for network community detection and the third experiment is designed to verify behavior of the algorithm and the time required to compute the algorithm. One of the major goals of the last experiment is to verify the behavior of the algorithm with respect to the performance of other popular methods exists in the literature in terms of standard modularity measures. Experiments are conducted over several real networks2 to compare the results (tables 7 and 8) of our algorithm with the state-of-the-art algorithms (table 1) available in the literature in terms of modularity most preferred by the researchers of the domain of network community detection.

In this experiment we have compared several algorithms for network community detection with our proposed algorithm developed using the nearest neighbor search in complex network. Experiment is performed on a large list of network data sets 2. Two version of the experiment is developed for comparison purpose based on modularity and time taken. The results are shown in the tables 7 and 8 respectively. In this experiment we have compared three groups of algorithms for network community detection with one based on nearest neighbor search, described above. Two version of the experiment is developed for comparison purpose based on modularity and time taken in seconds. Regarding the three groups of algorithms; first group contain algorithms based on semi-definite programming and the second group contain algorithms based on graph traversal approaches. For each group, we have taken the best value of modularity in table 7 among all the algorithms in the groups. Three different methods of network community detection are also considered for our comparison which indirectly uses the influence between the nodes in their algorithms. These methods are walktrap(WT), label propagation(LP) and geometric brownian motion(GBM). Three version of proposed algorithm is compared with other algorithms, the proposed algorithm based on exact nearest neighbor, approximated nearest neighbor computed using metric tree and approximate nearest neighbor computed using locality sensitive hashing.

3.5 Achievements

The results obtained in the first experiment justify that the proposed distance is more useful for complex network to extract the community structure compared to other measures of similarity. The results obtained in the second experiment justify that the proposed algorithm for community detection is very efficient compared to other existing methods in terms of modularity and time. The results obtained with our approach are very competitive with most of the well known algorithms in the literature and this is justified over the large collection of datasets. On the other hand, it can be observed that time (second) taken by our algorithm is quite less compared to other methods and justify the theoretical findings.

Table 5: Comparison of our approaches with other best methods in terms of modularity

Name	Spectral	SDP	GT	WT	LP	GBM	NN-search	M-tree	LSH
Facebook	0.4487	0.5464	0.5434	0.5117	0.5042	0.4742	0.5472	0.5450	0.5421
Gplus	0.2573	0.4047	0.3998	0.3528	0.3412	0.2963	0.4056	0.4041	0.4021
Twitter	0.3261	0.3706	0.3691	0.3545	0.3513	0.3375	0.3709	0.3692	0.3669
Epinions1	0.0280	0.1440	0.1401	0.1034	0.0940	0.0589	0.1447	0.1443	0.1437
LiveJournal1	0.0791	0.1455	0.1432	0.1220	0.1169	0.0966	0.1458	0.1450	0.1439
Pokec	0.0129	0.0294	0.0288	0.0235	0.0223	0.0172	0.0295	0.0292	0.0287
Slashdot0811	0.0038	0.0130	0.0127	0.0095	0.0090	0.0060	0.0125	0.0119	0.0117
Slashdot0922	0.0045	0.0176	0.0171	0.0127	0.0119	0.0078	0.0168	0.0165	0.0162
Friendster	0.0275	0.0536	0.0526	0.0443	0.0423	0.0343	0.0536	0.0531	0.0525
Orkut	0.0294	0.0689	0.0675	0.0549	0.0519	0.0398	0.0690	0.0685	0.0678
Youtube	0.0096	0.0934	0.0903	0.0640	0.0573	0.0319	0.0936	0.0934	0.0930
DBLP	0.4011	0.4214	0.4207	0.4136	0.4125	0.4060	0.4215	0.4196	0.4171
Arxiv-AstroPh	0.4174	0.5079	0.5045	0.4755	0.4688	0.4410	0.5081	0.5061	0.5035
web-Stanford	0.3595	0.3908	0.3896	0.3791	0.3772	0.3673	0.3908	0.3890	0.3866
Amazon0601	0.1768	0.2649	0.2615	0.2336	0.2269	0.1999	0.2650	0.2637	0.2621
P2P-Gnutella31	0.0009	0.0522	0.0503	0.0343	0.0301	0.0146	0.0523	0.0523	0.0523
RoadNet-CA	0.0212	0.1690	0.1633	0.1168	0.1053	0.0603	0.1692	0.1680	0.1664
Wiki-Vote	0.0266	0.2093	0.2023	0.1451	0.1306	0.0752	0.2095	0.2090	0.208

Table 6: Comparison of our approaches with other best methods in terms of time

Name	Spectral	SDP	GT	WT	LP	GBM	NN-search	M-tree	LSH
Facebook	6	7	11	13	7	8	6	4	1
Gplus	797	832	1342	1512	877	948	661	390	115
Twitter	462	485	786	886	509	554	398	235	68
Epinions1	411	419	667	749	452	475	292	174	56
LiveJournal1	1297	1332	2129	2394	1427	1514	969	576	179
Pokec	1281	1305	2075	2330	1410	1480	901	538	173
Slashdot0811	552	561	891	1000	608	636	382	228	74
Slashdot0922	561	570	906	1017	618	647	389	232	75
Friendster	2061	2105	3352	3766	2269	2390	1477	880	280
Orkut	1497	1529	2435	2736	1647	1735	1074	640	203
Youtube	829	844	1340	1505	913	957	578	345	111
DBLP	381	403	655	739	420	461	341	201	57
Arxiv-AstroPh	217	230	375	423	239	263	197	116	33
web-Stanford	498	525	852	960	549	600	437	258	74
Amazon0601	653	678	1089	1225	719	771	520	308	93
P2P-Gnutella31	182	184	293	328	200	209	124	74	24
RoadNet-CA	758	785	1261	1419	834	894	599	355	107
Wiki-Vote	54	55	88	99	59	63	39	23	7

4 Low-rank approximation for very large networks

In this chapter we have proposed an algorithm to reduce the dimensionality of the datasets such that after applying partitioning on reduced network we get almost same results as with the original network via Nystrom sampling and random sketches. For evaluation of the proposed approaches on complex network we applied it in community detection problem. The results obtained using our methods are very competitive with most of the well known algorithms exists in the literature and this is verified on collection of real networks. On the other-hand, it can be observed that time taken by our algorithm is quite less compared to popular methods.

4.1. Low-rank approximation methods

In this work we will explore the two low rank approximation methods; Nystrom sampling and random sketching and they are described below.

4.2. Nystrom approximation

The Nystrom methods approximate any symmetric positive semi definite (SPSD) matrix in terms of a subset of its columns. Specifically, given an m × m SPSD matrix A, they require sampling c(< m) columns of A to construct an m × c matrix C. We always assume that C consists of the first c columns of A without loss of generality. We partition A and C as

$$A = \begin{pmatrix} W & A_{21}^T \\ A_{21} & A_{22} \end{pmatrix} \; and \; C = \begin{pmatrix} W \\ A_{21} \end{pmatrix}$$

where W and A_{21} are of size c × c and (m – c) × c respectively [32]

The standard Nystrom approximation of A is defined as:

$$A_c^{nys} = \begin{pmatrix} W & A_{21}^T \\ A_{21} & A_{21}W\,A_{21}^T \end{pmatrix}$$

Since the running time complexity of SVD on W is in O(kl2) and matrix multiplication with C takes O(kln), the total complexity of the Nystrom approximation computation is in O(kln) [2,17].

4.3 Random Sketching

In random sketch, firstly relation is modeled as defining a vector or matrix, and then the sketch is formed by multiplying the data by a random vector. Sketch vector forms the synopsis of the entire data, which is much smaller than the original data. Data mining algorithms can now be applied on the sketch vector.

Johnson-Lindenstrauss Lemma 1: For any set of n points S ∈ Rd, there is a (1 +€)-distortion embedding of X into R^k(k < d), for k = O(log2(n)/e^2)

Johnson-Lindenstrauss Lemma 2: There is a distribution over random linear mappings A : $R^d \rightarrow R^k$, (k < d) such that for any vector x we have ||Ax|| = (1 ±)||x|| with probability 1 − e^{-Cke2}

The Johnson Linderstrauss theorems [30] for sketches ensure the preservation of distance during lower dimensional approximation and provide minimum lower bound of approximation.

4.4 Experiments on low-rank approximation

We performed many of experiments to test the performance of low rank approximation based community detection method for complex network over several real networks2. Objective of the experiment is to verify behavior of the algorithm and the time required to compute the algorithm. One of the major goals of the experiment is to verify the behavior of the algorithm with respect to the performance of other popular methods exists in the literature with respect to the modularity. Experiment for comparison: In this experiment we have compared several algorithms for network community detection with our proposed algorithm developed using nearest neighbor search in complex network. Experiment is performed on a large list of network data sets. Two version of the experiment is developed for comparison purpose based on modularity. Experiment on performance and time: In this experiment we have evaluated our algorithm for performance on the network collection. We have evaluated the time taken by our algorithm on different size of networks and is shown in the table below.

Table 7: Comparison of low-rank approximation with other methods in terms of modularity

Name	Spectral	SDP	GT	k-center	Nystrom	R-sketches
Facebook	0.4487	0.5464	0.5434	0.5472	0.5450	0.5421
Gplus	0.2573	0.4047	0.3998	0.4056	0.4041	0.4021
Twitter	0.3261	0.3706	0.3691	0.3709	0.3692	0.3669
Epinions1	0.0280	0.1440	0.1401	0.1447	0.1443	0.1437
LiveJournal1	0.0791	0.1455	0.1432	0.1458	0.1450	0.1439
Pokec	0.0129	0.0294	0.0288	0.0295	0.0292	0.0287
Slashdot0811	0.0038	0.0130	0.0127	0.0131	0.0129	0.0127
Slashdot0922	0.0045	0.0176	0.0171	0.0176	0.0174	0.0172
Friendster	0.0275	0.0536	0.0526	0.0536	0.0531	0.0525
Orkut	0.0294	0.0689	0.0675	0.0690	0.0685	0.0678
Youtube	0.0096	0.0934	0.0903	0.0936	0.0934	0.0930
DBLP	0.4011	0.4214	0.4207	0.4215	0.4196	0.4171
Arxiv-AstroPh	0.4174	0.5079	0.5045	0.5081	0.5061	0.5035
web-Stanford	0.3595	0.3908	0.3896	0.3908	0.3890	0.3866
Amazon0601	0.1768	0.2649	0.2615	0.2650	0.2637	0.2621
P2P-Gnutella31	0.0009	0.0522	0.0503	0.0523	0.0523	0.0523
RoadNet-CA	0.0212	0.1690	0.1633	0.1692	0.1680	0.1664
Wiki-Vote	0.0266	0.2093	0.2023	0.2095	0.2090	0.2083

Table 8: Comparison of low-rank approximation with other methods in terms of time

Name	Spectral	SDP	GT	k-center	Nystrom	R-sketches
Facebook	6	7	11	6	4	1
Gplus	797	832	1342	661	390	115
Twitter	462	485	786	398	235	68
Epinions1	411	419	667	292	174	56
LiveJournal1	1297	1332	2129	969	576	179
Pokec	1281	1305	2075	901	538	173
Slashdot0811	552	561	891	382	228	74
Slashdot0922	561	570	906	389	232	75
Friendster	2061	2105	3352	1477	880	280
Orkut	1497	1529	2435	1074	640	203
Youtube	829	844	1340	578	345	111
DBLP	381	403	655	341	201	57
Arxiv-AstroPh	217	230	375	197	116	33
web-Stanford	498	525	852	437	258	74
Amazon0601	653	678	1089	520	308	93
P2P-Gnutella31	182	184	293	124	74	24
RoadNet-CA	758	785	1261	599	355	107
Wiki-Vote	54	55	88	39	23	7

4.5 Achievements

Low-rank methods are very fast to compute communities from massive networks by preserving the approximated results with moderate lower bound

5 Optimal estimation for community detection

In this chapter, we compare communities of complex networks determined by several algorithms using Rough Sets theory and tried to combine them to obtain an optimally community structure of the network. A community detection algorithm determines a partition of the network. Given several partitions of same network obtained from the output of different community detection algorithms, we tried to generate a new partition of same size with better modularity than all. The comparison of partitions is done using Rough Sets by assigning the minimum discernibility of the new partition with all others.

5.1 Theoretical developments

Rough set (developed by Polish mathematician Zdzislaw I. Pawlak in 1991) is a formal approximation of a concept in terms of a pair of sets, the lower and the upper approximation derived from the available instances [45]. Formally, an information system is defined as I = (U, A, V, f), where U is a nonempty and finite set of objects, called the universe; A is a nonempty and finite set of attributes; V is the union of attribute domains, i.e., V = ∪Va, where Va denotes the domain for each attribute a ∈ A; and f is an information function which associates a unique value of each attribute with every object belonging to U. An arbitrary attribute set B ⊆ A determines a binary relation IND(B), called in discernibility relation. IND(B) = {< x, y >∈ U × U|∀b ∈ B, f(x, b) = f(y, b)} Equivalence partition denoted as U/B and blocks are written as $[x]_B$ = {y ∈ U| < x, y >∈ IND(B)} Let C ⊆ U be a concept, the lower and upper approximations with respect to B ⊆ A are denoted as

$$\underline{B(X)} = \{x \in U | [x]_B \subseteq X\} \ \& \ \overline{B(X)} = \{x \in U | [x]_B \cap X \neq \Phi\}$$

Algorithm 5 Optimal community detection using rough discernibility

Require: $G = (V, E)$, $\Pi_1, \Pi_2, \Pi_3, \Pi_4, \ldots, \Pi_k$
Ensure: Π_o - the optimal community
1: Let Π_s be the super-partition of $\Pi_1, \Pi_2, \Pi_3, \Pi_4, \ldots, \Pi_k$
2: Find $\Pi_x \subset \Pi_s$ s.t. $min_{i=1}^{k}\{|\Pi_i|\} \leq |\Pi_x| \leq max_{i=1}^{k}\{|\Pi_i|\}$
3: $\Pi_o = \Pi_x$ with minimum discernibility
4: **return** Π_o

Theorem 5.1: Newly obtained meta-partition has better community structure than its contributing partitions in terms of modularity.

Theorem 5.2: Newly obtained meta-partition is estimation to the optimal community structure of the network and converges towards optimal community structure with increasing number of input partitions.

Theorem 5.3: Newly obtained meta-partition converges towards optimal community structure faster when input partitions have more mutual diversity.

5.2. Experiments on optimal communities

In this experiment, we combine the communities of complex networks determined by several algorithms with Rough Sets theory to obtain a new optimal community structure of the network. We obtained improved results of community detection in less time by combining results of approximate algorithms and are shown in the tables 9 & 10.

Table 9: Experiment on optimal community structure estimation

	Exact		LSH		Nystrom		Sketches		Combined	
Networks	Mod	T(s)	Mod	T(s)	Mod	T(s)	Mod	T(s)	Mod	T(s)
Facebook	0.5472	6	0.5421	1	0.5405	1	0.5397	1	0.5526	2
Gplus	0.4056	661	0.4021	115	0.4010	92	0.4004	100	0.4061	77
Twitter	0.3709	398	0.3669	68	0.3656	54	0.3650	59	0.3763	45
Epinions1	0.1447	292	0.1437	56	0.1434	44	0.1432	48	0.1419	37
LiveJournal1	0.1458	969	0.1439	179	0.1433	143	0.1430	156	0.1466	119
Pokec	0.0295	901	0.0287	173	0.0285	138	0.0283	150	0.0302	115
Slashdot0811	0.0125	382	0.0117	74	0.0114	59	0.0113	64	0.0143	49
Slashdot0922	0.0168	389	0.0162	75	0.0160	60	0.0159	65	0.0182	50
Friendster	0.0536	1477	0.0525	280	0.0522	224	0.0520	243	0.0546	187
Orkut	0.0690	1074	0.0678	203	0.0674	163	0.0672	177	0.0697	136
Youtube	0.0936	578	0.0930	111	0.0928	89	0.0927	97	0.0913	74
DBLP	0.4215	341	0.4171	57	0.4157	46	0.4150	50	0.4286	38
Arxiv-AstroPh	0.5081	197	0.5035	33	0.5021	26	0.5013	28	0.5128	22
web-Stanford	0.3908	437	0.3866	74	0.3853	59	0.3846	64	0.3972	49
Amazon0601	0.2650	520	0.2621	93	0.2612	74	0.2607	81	0.2668	62
P2P-Gnutella31	0.0523	124	0.0523	24	0.0520	19	0.0520	21	0.0507	16
RoadNet-CA	0.1692	599	0.1664	107	0.1655	86	0.1651	93	0.1683	71
Wiki-Vote	0.2095	39	0.2083	7	0.2079	6	0.2077	6	0.2045	5

Table 10: Summary of results

Netrorks	Exact(M)	Exact(T)	RO(M)	RO(T)	Modularity Gain	Time Gain
Facebook	0.5472	6	0.5526	2	0.0054	5
Gplus	0.4056	661	0.4061	77	0.0005	584
Twitter	0.3709	398	0.3763	45	0.0054	353
Epinions1	0.1447	292	0.1419	37	-0.0028	255
LiveJournal1	0.1458	969	0.1466	119	0.0008	850
Pokec	0.0295	901	0.0302	115	0.0007	786
Slashdot0811	0.0125	382	0.0143	49	0.0018	333
Slashdot0922	0.0168	389	0.0182	50	0.0014	339
Friendster	0.0536	1477	0.0546	187	0.0010	1290
Orkut	0.069	1074	0.0697	136	0.0007	939
Youtube	0.0936	578	0.0913	74	-0.0023	504
DBLP	0.4215	341	0.4286	38	0.0071	303
Arxiv-AstroPh	0.5081	197	0.5128	22	0.0047	176
web-Stanford	0.3908	437	0.3972	49	0.0064	388
Amazon0601	0.265	520	0.2668	62	0.0018	458
P2P-Gnutella31	0.0523	124	0.0507	16	-0.0016	108
RoadNet-CA	0.1692	599	0.1683	71	-0.0009	528
Wiki-Vote	0.2095	39	0.2045	5	-0.0051	34

5.3.Achievements

The optimal combination of the communities obtained from several algorithms in-crease the modularity resultant community structure. This combination method is computationally very efficient when approximate algorithms are used.

6 CONCLUSION

In this work the fast and optimal algorithms for community detection on large net-work is described and analyzed. We demonstrated and analyzed the new approaches to network community detection via metric space induced by the complex network. The interesting problem of the nearest neighbor within the nodes of a complex net-works are studied and applied for community detection. We have used geometric framework for network community detection instead of traditional graph theoretic approach or spectral methods. We presented the efficient computation of network community detection using low rank approximation. Our techniques can be applied for quick analysis of very large complex network. Finally the theoretical upper bound of network community detection algorithms is analyzed using the notion of data complexity and tried to estimate the optimal bound by heterogeneous combination of communities.

7 REFERENCES

[1] Karim T. Abou-Moustafa, Dale Schuurmans, and Frank P. Ferrie. Learning a metric space for neighbourhood topology estimation: Application to manifold learning. In Asian Conference on Machine Learning, ACML 2013, Canberra, ACT, Australia, November 13-15, 2013, pages 341–356, 2013.

[2] Dimitris Achlioptas and Frank Mcsherry. Fast computation of low-rank matrix approximations. J. ACM, 54(2), 2007.

[3] Leman Akoglu, Rohit Khandekar, Vibhore Kumar, Srinivasan Parthasarathy, Deepak Rajan, and Kun-Lung Wu. Fast nearest neighbor search on large time-evolving graphs. In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8724, ECML PKDD 2014, pages 17–33, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[4] Rodrigo Aldecoa and Ignacio Marin. Exploring the limits of community detection strategies in complex networks. Sci. Rep., 3, 2013.

[5] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for ap-proximate nearest neighbor in high dimensions. Commun. ACM, 51(1):117–122, 2008.

[6] James P. Bagrow and Erik M. Bollt. Local method for detecting communities. Phys. Rev. E, 72:046108, 2005.

[7] Andrea Capocci, Vito D. P. Servedio, Guido Caldarelli, and Francesca Colaiori. Detecting communities in large networks. Physica A: Statistical Mechanics and its Applications, 352(2-4):669–676, 2004.

[8] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC ’02, pages 380–388, New York, NY, USA, 2002. ACM.

[9] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In Proceedings of the 23rd In-ternational Conference on Very Large Data Bases (VLDB'97), pages 426–435, Athens, Greece, 1997. Morgan Kaufmann Publishers, Inc.

[10] Paolo Ciaccia, Marco Patella, and Pavel Zezula. A cost model for similarity queries in metric spaces. In Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97), pages 59– 68, Seattle, WA, 1998. ACM Press.

[11] Aaron Clauset, M. E. J. Newman, , and Cristopher Moore. Finding community structure in very large networks. Physical Review E, pages 1– 6, 2004.

[12] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for com-munity discovery methods in complex networks. Statistical Analysis and Data Mining, 4(5):512–546, 2011.

[13] Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimen-sional manifolds. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08, pages 537–546, New York, NY, USA, 2008. ACM.

[14] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, SCG '04, pages 253– 262, New York, NY, USA, 2004. ACM.

[15] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Mixing local and global information for community detection in large networks.

J. Comput. Syst. Sci., 80(1):72–87, 2014.

[16] David Deritei, Zsolt I Lazar, Istvan Papp, Ferenc Jarai-Szabo, Robert Sumi, Levente Varga, Erzsebet Ravasz Regan, and Maria Ercsey-Ravasz. Community detection by graph voronoi diagrams. New Journal of Physics, 16(6):063007, 2014.

[17] Petros Drineas and Michael W. Mahoney. On the nystrom method for approxi-mating a gram matrix for improved kernel-based learning. J. Mach. Learn. Res., 6:2153–2175, 2005.

[18] J. Duch and A. Arenas. Community detection in complex networks using ex-tremal optimization. Physical Review E, 72:027104, 2005.

[19] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. PNAS, 99(9):5825–5829, 2002.

[20] Santo Fortunato. Community detection in graphs. Physics Reports, 486(3-5):75 – 174, 2010.

[21] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find commu-nity structures based on information centrality. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), 70(5):056104, 2004.

[22] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[23] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[24] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12):7821–7826, 2002.

[25] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance.

Theor. Comput. Sci., 38:293–306, 1985.

[26] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. Nature, 433(7028):895–900, 2005.

[27] Piotr Indyk. A sublinear time approximation scheme for clustering in metric spaces. In Proc. 40th IEEE FOCS, pages 154–159, 2000.

[28] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[29] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC ’98, pages 604–613, New York, NY, USA, 1998. ACM.

[30] William Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In Conference in modern analysis and probability (New Haven, Conn., 1982), volume 26 of Contemporary Mathematics, pages 189–206. American Mathematical Society, 1984.

[31] Alexis Joly and Olivier Buisson. A posteriori multi-probe locality sensitive hash-ing. In Abdulmotaleb El-Saddik, Son Vuong, Carsten Griwodz, Alberto Del Bimbo, K. SelÃğuk Candan, and Alejandro Jaimes, editors, ACM Multimedia, pages 209–218. ACM, 2008.

[32] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nystrom method. J. Mach. Learn. Res., 13(1):981–1006, 2012.

[33] Conrad Lee and PÃądraig Cunningham. Community detection: effective evalu-ation on large social networks. Journal of Complex Networks, 2013.

[34] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, 2008.

[35] Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical comparison of algorithms for network community detection. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, WWW, pages 631–640. ACM, 2010.

[36] Ting Liu, Andrew W. Moore, Er Gray, and Ke Yang. An investigation of prac-tical approximate nearest neighbor algorithms. In NIPS2004, pages 825–832. MIT Press, 2004.

[37] Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. SIAM J. Discrete Math., 21(4):930–935, 2007.

[38] M E Newman. Modularity and community structure in networks. Proc Natl Acad Sci U S A, 103(23):8577–8582, 2006.

[39] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review, E 69(026113), 2004.

[40] M.E.J. Newman. Fast algorithm for detecting community structure in networks.

Physical Review E, 69, 2003.

[41] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping com-munity structure of complex networks in nature and society. Nature, 435:814– 818, 2005.

[42] Gang Pan, Wangsheng Zhang, Zhaohui Wu, and Shijian Li. Online community detection for large complex networks. PLoS ONE, 9:e102799, 2014.

[43] Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, SODA ’06, pages 1186–1195, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics.

[44] Lo¨ıc Paulev´e, Herv´ J´egou, and Laurent Amsaleg. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. Pattern Recogn. Lett., 31(11):1348–1358, 2010.

[45] Zdzislaw Pawlak. Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[46] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. J. of Graph Alg. and App., 10:284–293, 2004.

[47] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences, 101(9):2658, 2004.

[48] Usha Nandini Raghavan, R´eka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E, 76:036106, 2007.

[49] J Reichardt and S Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. Phys Rev Lett, 93:218701, 2004.

[50] Peter Ronhovde and Zohar Nussinov. Multiresolution community detection for megascale networks by information-based replica correlations. Phys. Rev. E, 80:016109, 2009.

[51] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

[52] Enrique Vidal Ruiz. An algorithm for finding nearest neighbours in (approx-imately) constant average time. Pattern Recognition Letters, 4(3):145 – 157, 1986.

[53] Satu Elisa Schaeffer. Graph clustering. Computer Science Review, 1(1):27 – 64, 2007.

[54] Jeffrey K. Uhlmann. Satisfying general proximity / similarity queries with metric trees. Information Processing Letters, 40(4):175 – 179, 1991.

[55] S. van Dongen. A Cluster Algorithm For Graphs. Technical Report INS-R 0010, CWI, Amsterdam, the Netherlands, 2000.

[56] F. Wu and B.A. Huberman. Finding communities in linear time: a physics approach. The European Physical Journal B - Condensed Matter and Complex Systems, 38(2):331–338, 2004.

[57] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu, editors, ICDM, pages 745–754. IEEE Computer Society, 2012.

[58] Mina Zarei, Keivan Aghababaei Samani, and Gholam Reza Omidi. Com-plex eigenvectors of network matrices give better insight into the commu-nity structure. Journal of Statistical Mechanics: Theory and Experiment, 2009(10):P10018, 2009.

[59] Haijun Zhou and Reinhard Lipowsky. Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcom-munities. In Marian Bubak, G. Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, International Conference on Computational Science, volume 3038 of Lecture Notes in Computer Science, pages 1062–1069. Springer, 2004.

8 LIST OF PUBLICATIONS

Journal Publications

[1] Suman Saha, S. P. Ghrera, Low Rank Approximation for Community Detection in Large Complex Networks, Far East Journal of Electronics and Communications,2016. (Accepted) [Scopus]

[2] Suman Saha, S.P. Ghrera, Nearest Neighbor search in the Metric Space of Com-plex Network for Community Detection, Information, 8 (Special Issue: Online Social Networks and Implications), 2016. [Scopus]

[3] Suman Saha, S.P. Ghrera, Network Community Detection on Metric Space, Algorithms, 8 (Special Issue: Clustering Algorithms), 680-696, 2015. [Scopus, Web of Science]

[4] Mohit Sharma, Suman Saha, An Efficient Round robin Minimum Spanning Tree Algorithm in MapReduce Framework, INROADS-An International Journal of Jaipur National University 3 (1s), 106-110, 2014.

[5] C.P. Singh, Suman Saha, S.K. Saurabh, Link Analysis to Visualize a Web Graph, International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 3, p247, 2012.

Conference Proceedings

[6] Rawat, A., Saha, S., & Ghrera, S. P. (2015, December). Time efficient ranking system on map reduce framework. In 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 496-501). IEEE.. [Scopus]

[7] Patel, L. S., Sana, S., & Ghrera, S. P. (2015, December). Efficient Nystrom method for low rank approximation and error analysis. In 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 536-542). IEEE. [Scopus]

[8] Sharma, M., & Saha, S. (2014, September). Graph based approach for minimum multicollinearity highly accurate regression model explaining maximum variability. In Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference- (pp. 304-308). IEEE. [Scopus]

[9] K Arvind, J Suchi, S Suman, Reduction in Searching Time of Inverted Index using Bloom Filter, ERCICA-2014. [Scopus]

[10] M Sharma, S Suman, Robust Best Regressor Model with Minimum Multi-collinearity, I2CT-2014, Pune,India.

[11] Gupta, P., Agnihotri, S., & Saha, S. (2013). Approximate data mining using sketches for massive data. Procedia Technology, 10, 781-787.. [Scopus]